

NAME

CURLOPT_SEEKFUNCTION – user callback for seeking in input stream

SYNOPSIS

```
#include <curl/curl.h>
```

```
/* These are the return codes for the seek callbacks */
```

```
#define CURL_SEEKFUNC_OK    0
```

```
#define CURL_SEEKFUNC_FAIL  1 /* fail the entire transfer */
```

```
#define CURL_SEEKFUNC_CANTSEEK 2 /* tell libcurl seeking can't be done, so
libcurl might try other means instead */
```

```
int seek_callback(void *userp, curl_off_t offset, int origin);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SEEKFUNCTION, seek_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

This function gets called by libcurl to seek to a certain position in the input stream and can be used to fast forward a file in a resumed upload (instead of reading all uploaded bytes with the normal read function/callback). It is also called to rewind a stream when doing a HTTP PUT or POST with a multi-pass authentication method. The function shall work like `fseek(3)` or `lseek(3)` and it gets `SEEK_SET`, `SEEK_CUR` or `SEEK_END` as argument for *origin*, although libcurl currently only passes `SEEK_SET`.

userp is the pointer you set with `CURLOPT_SEEKDATA(3)`.

The callback function must return `CURL_SEEKFUNC_OK` on success, `CURL_SEEKFUNC_FAIL` to cause the upload operation to fail or `CURL_SEEKFUNC_CANTSEEK` to indicate that while the seek failed, libcurl is free to work around the problem if possible. The latter can sometimes be done by instead reading from the input or similar.

If you forward the input arguments directly to `fseek(3)` or `lseek(3)`, note that the data type for *offset* is not the same as defined for `curl_off_t` on many systems!

DEFAULT

By default, this is NULL and unused.

PROTOCOLS

HTTP, FTP, SFTP

EXAMPLE

TODO

AVAILABILITY

Added in 7.18.0

RETURN VALUE

Returns `CURLE_OK` if the option is supported, and `CURLE_UNKNOWN_OPTION` if not.

SEE ALSO

`CURLOPT_SEEKDATA(3)`, `CURLOPT_IOCTLFUNCTION(3)`,